International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13, Issue 4, April-2024, Impact Factor: 8.285
Presented at "ICRETETM-2024", Organized by GSMCOE, Pune, on 22nd - 23rd April 2024

WebApp Vulnerability Scanner

Om Namade[1], Ketaki Ganesh Katre[2], Shantanu Kulkarni[3], Ganesh Rupanwar[4]

[1]FirstAuthor, Department of Information Technology, Genba Sopanrao Moze, Balewadi, Pune, India
[2]Guide, Department of Information Technology, Genba Sopanrao Moze, Balewadi, Pune, India
[3,4]Third author & Fourth Author, Department of Information Technology, Genba Sopanrao Moze, Balewadi, Pune, India

ABSTRACT

Web Application Vulnerability Scanners (WAVS) play a pivotal role in modern cybersecurity practices, providing essential tools for identifying and mitigating potential threats within web applications. This abstract delves into the significance of WAVS in safeguarding online assets, exploring their methodologies, critical features, and the impact they have on enhancing overall security postures. By employing techniques such as black-box testing, input fuzzing, and pattern recognition, these scanners systematically evaluate web applications, pinpointing vulnerabilities like SQL injections, Cross-Site Scripting (XSS), and authentication weaknesses. Comprehensive reporting and continuous monitoring features ensure that organizations can respond promptly to emerging threats. Collaboration with penetration testing further strengthens security evaluations. Understanding the nuances of WAVS is fundamental for organizations striving to maintain robust defences against everevolving cyber threats, thereby ensuring the integrity, confidentiality, and availability of their web-based services and data.

Keywords: Scanning and Mapping, Web Services Testing, Penetration Testing, Analysis and Reporting

INTRODUCTION

A Web App Vulnerability Scanner is crucial for cybersecurity, evaluating web app security using automated methods. It crawls, maps, and tests for vulnerabilities like SQL injections, XSS, and authentication flaws. This analysis helps organizations proactively address weaknesses, safeguarding data integrity and user confidentiality against cyber threats.

[1] Scanning and Mapping:

Scanning refers to the process where a vulnerability scanner systematically navigates through a web application, exploring its various pages, links, and inputs and detects the vulnerability using payloads.
The purpose of vulnerability scanning is to proactively identify weaknesses, flaws, and potential security risks within an organization's network, systems, and applications. This process involves using specialized software tools to systematically scan and analyse various components of an IT infrastructure to pinpoint vulnerabilities that could be exploited by attackers.

[2] Web Services Testing:

- Web Services: Web services are software systems designed to allow for interoperable interaction over a network. They are used for communication between different applications, often through APIs (Application Programming Interfaces) using standard web protocols such as HTTP and XML.
- Critical Component: In modern applications, web services are a critical component, often serving as the backbone for data exchange and functionality.
- Security Concerns: Due to their widespread use, web services are vulnerable to various attacks, making their security testing crucial to ensure data integrity and confidentiality.

[3] Penetration Testing:

- Penetration Testing: Penetration testing, often referred to as ethical hacking, is a simulated cyber-attack on a computer system, network, or web application to evaluate its security strength. In the context of web applications, penetration testing involves active analysis, real-time exploitation, and testing of vulnerabilities to assess the application's security posture.

LITERATURE REVIEW

Research on vulnerability scanners examines methodologies, effectiveness, and the balance between automation and human expertise. Scholars evaluate their accuracy in detecting known vulnerabilities and their application to specific environments like web applications and network infrastructure. Furthermore, studies explore how vulnerability scanners impact cybersecurity practices, including risk management and compliance adherence. Overall, the literature underscores the crucial role of vulnerability scanners in identifying and mitigating security threats in diverse digital environments.

METHODOLOGY:

- In V-WAVS we have used various methods to scan for web application vulnerabilities. By using the mean of cyber security and python programming combined we have created a high-end vulnerability scanner.

1) Collection and processing of the Data: - From the user we collect the **Target URL(s**) or the **IP Address(s)** of the web application on which we want to perform the vulnerability scan.

2) Processing the collected data: - The collected data is proceeded to the programming logics in order to scan for vulnerabilities. The payloads and the hacking techniques are pre-defined in the programming logic in which the URL or IP go through.

3) Scanning and Detecting: - The programming logic and the payloads automatically detects the vulnerabilities in the web applicating as per the payload. Some commonly found vulnerabilities are – SQLi, XSS, Insecure File.

4) Reporting and Solution: - Found vulnerabilities are shown in the interface of the system and the solution for them is also shown accordingly as per the detection.

5) Exploitation: This is a **restricted** feature of this application which can be used on some terms and conditions by agree to that the user can exploit the web application. By exploiting the web app user can perform the attack as per thevulnerabilities accordingly.

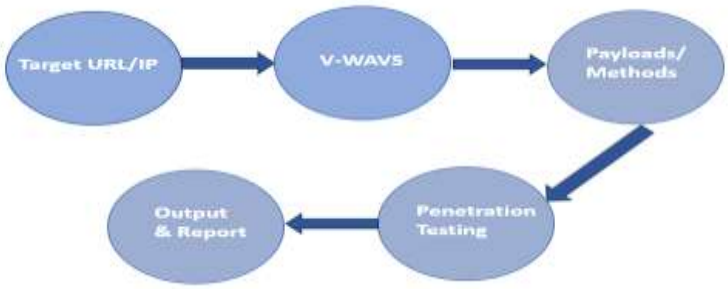


Figure 1:Methodology

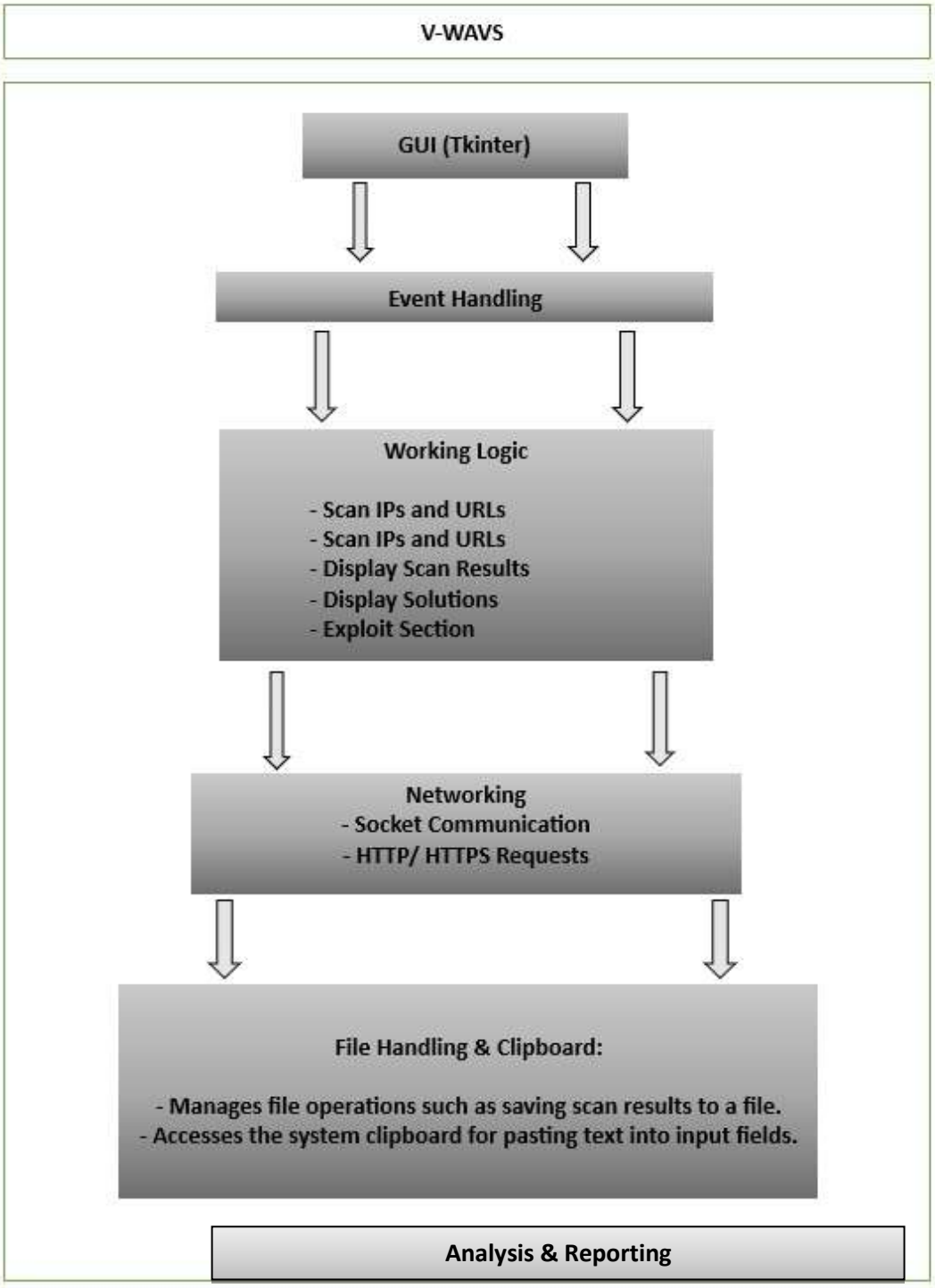


Figure 2:System Architecture

ARCHITECTURE

1. GUI (Graphical User Interface):

- The GUI component utilizes the Tkinter library to create a visually appealing and interactive interface for users.
- It includes various widgets such as buttons, entry fields, labels, and text areas to facilitate user input and display information.
- Responsible for presenting scan results, input fields for IP addresses and URLs, and options for scan types.
- Provides a user-friendly environment for initiating scans, viewing results, and interacting with the application.

2. Event Handling:

- Event Handling manages user interactions within the GUI, interpreting user inputs and triggering appropriate actions.
- It captures events like button clicks, dropdown selections, and keyboard inputs, ensuring responsiveness to user actions.
- Orchestrates the flow of events within the application, directing them to the relevant components for processing.
- Facilitates seamless interaction between users and the application, enhancing usability and user experience.

3. Working Logic:

- The Working Logic component embodies the core functionality of the application, implementing the logic behind scanning IPs and URLs.
- Responsible for orchestrating scan processes, handling custom scan types and vulnerabilities, and processing scan results.
- Interacts with the networking module to perform IP and URL scanning, as well as with the file handling component for saving scan results.
- Manages the application's state and data flow, ensuring accurate processing and presentation of scan information.

4. Networking:

- The Networking component handles network communication required for scanning IPs and URLs.
- Utilizes sockets for establishing connections and exchanging data with remote hosts during the scanning process.
- Makes HTTP requests using the requests library for retrieving content from web URLs.
- Ensures reliable communication with external systems, facilitating the retrieval of scan data and detection of vulnerabilities.

5. File Handling & Clipboard:

- The File Handling & Clipboard component manages file operations and interacts with the system clipboard.
- Handles operations such as saving scan results to a file and accessing clipboard data for pasting text into input fields.
- Ensures seamless integration with system resources, facilitating data exchange and persistence within the application.
- Enhances user convenience by enabling easy access to scan results and enabling efficient data input via clipboard interactions.

6. Exploit:

- The Exploit component provides functionality for exploiting vulnerabilities found during the scanning process.
- Users should exercise caution and use this component responsibly, preferably on their own websites or with proper authorization.
- It may include features for automated exploitation or manual intervention, depending on the nature of the vulnerabilities detected.
- Enables users to take appropriate actions to mitigate vulnerabilities and enhance the security of their systems.

7.Analysis and Reporting

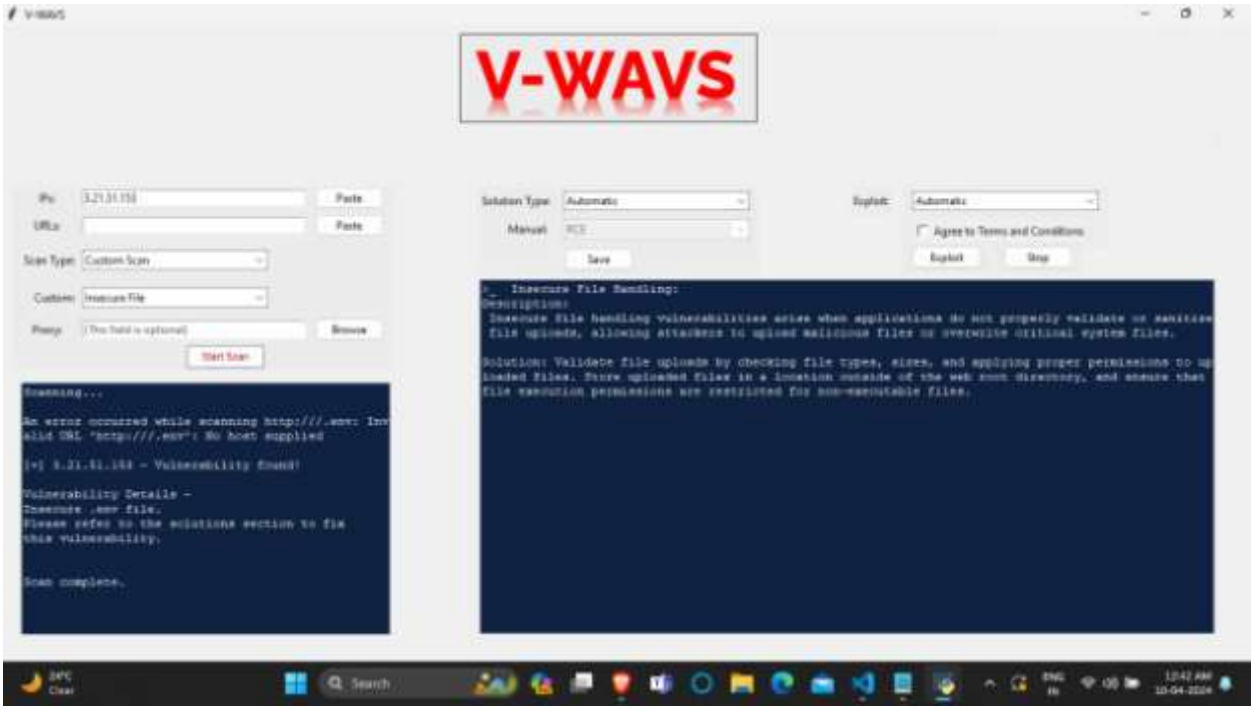
- Automated Vulnerability Identification: Web application vulnerability scanners automatically identify vulnerabilities through various testing techniques such as crawling, input fuzzing, and pattern recognition.

- Correlation and Context: Scanners correlate collected data, considering the context in which vulnerabilities were discovered, aiding in the prioritization of critical issues.
- Comprehensive Reports: Vulnerability scanners generate detailed reports, including information about each identified vulnerability, its location, potential impact, and remediation recommendations.

Each component plays a crucial role in the architecture of the V-WAVS application, contributing to its functionality, usability, and security. Together, they form a cohesive system that enables users to scan IPs and URLs effectively, analyze vulnerabilities, and take appropriate actions based on the scan results, including exploiting vulnerabilities if necessary and authorized.

Figure 3:Working Process

Result



CONCLUSION

A web application vulnerability scanner is a crucial tool for identifying and mitigating security risks. By automating the detection of vulnerabilities, it helps safeguard against potential threats and breaches. Its systematic approach streamlines the

process of securing web applications, enhancing overall cybersecurity posture. Embracing such tools is paramount in maintaining the integrity and resilience of online platforms in an ever-evolving threat landscape.

REFERENCES

[1]. OWASP. (2021). OWASP Application Security Verification Standard. Retrieved from https://owasp.org/www-project-application-securityverification-standard/
[2]. NIST. (2021). National Vulnerability Database (NVD). Retrieved from https://nvd.nist.gov/
[3]. Moore, D., & Shannon, C. (2005). Code Red: A Case Study on the Spread and Victims of an Internet Worm. Proceedings of the 2002 ACM Workshop on Rapid Malcode (WORM), 1-13. [Link](https://www.researchgate.net/publication/221036800_Code_red_A_case_study_on_the_spread_and_victims_of_an_Internet_worm/)
[4]. Shaw, J. (2020). Python for Penetration Testing: A Comprehensive Guide. Packt Publishing.
[5]. McGrew, D., & Hieb, J. L. (2018). Network Scanning with Python: Discover the power of scripting with Python and Scapy to advance your network scanning skills. Packt Publishing.
[6]. Tso, R., Nguyen, L., & Okumura, M. (2019). Evaluation of Open Source Network Vulnerability Scanners. In Proceedings of the 14th International Conference on Cyber Warfare and Security (ICCWS 2019), 540-548. [Link](https://www.researchgate.net/publication/334027881_Evaluation_of_Open_Source_Network_Vulnerability_Scanners